**תכנות מונחה עצמים מתקדם**

**עבודת הגשה מס 4'**

**להגשה עד ה- 03/06/2019 ב- 23:55**

**נא לקרוא את כל המסמך עד סופו לפני תחילת העבודה!**

**דגשים להגשה**

* ניתן להגיש עבודה זו בזוגות – רק אחד מהסטודנטים יגיש את העבודה במודל. בתיעוד של קובץ יש לציין שם,ו ת.ז של מגיש\ים, בתוך תיעוד הjavadoc
* בדיקת העבודה מתבצעת גם ידנית ולכן חשוב מאוד להקפיד על תיעוד!
* לכל שאלה אנא עברו על מסמך ה[FAQ](https://docs.google.com/document/d/1rq4r5ogEOLsVdRQ3l1WSAR236tPd2hD4btRqvRBDaOk/edit?usp=sharing) ובדקו האם היא נענתה כבר. במידה ולא, ניתן להוסיף הצעת עריכה בקובץ או לפנות לאחראית התרגיל - תמר שרוט - tammarm@gmail.com
* **על כל פניה** להכיל את פרטי הסטודנט המלאים (2 הסטודנטים אם מדובר בזוג) כולל שם ות.ז ושם המתרגל.
* חובה לתעד כל קובץ, מחלקה ופונקציה ע"י javaDoc.  
  ניתן להיעזר [בתיעוד באתר oracle](http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html) או בקבצים הרלוונטיים במודל.

**דגשים לעבודה זו**

* בעבודה זאת מתרגלים תבניות עיצוב שכל מטרתם להעלות את רמת העיצוב של הקוד. ולכן יש לדאוג לעיצוב סביר והגיוני. אין לבצע מניפולציות שיסבכו את הקוד רק על מנת לעמוד ביעדים. חישבו כיצד טוב ביותר לעצב את הקוד מחדש (אם יש צורך).
* ניתן להוסיף מחלקות עזר. וכן ניתן להוסיף שדות ומתודות למחלקות לפי צורך.
* על כל העבודה להיות פרויקט יחיד המחולק לpackages לפי המטלות.
* על כל השדות בכל המחלקות להיות פרטיים בלבד.
* על כל הSettres להיות בוליאניים, לבדוק תקינות נתונים (אם הוגדרה) ולהחזיר האם בוצעה השמה. Settres שרק למחלקות בנות אמורה להיות גישה אליהן, יהיו protected.
* באופן כללי כל מתודה שלא מחזירה ערך חישובי צריכה להיות בוליאנית (הצלחה או כישלון).
* לכל מחלקה יש לממש: בנאי ומתודת toString וכן מתודות setו-get לפי הצורך בלבד.
* עליכם להימנע משכפול קוד ככל שניתן.

בתרגיל זה עליכם לוודא שהמערכת שלכם מממשת לפחות את תבניות העיצוב שיפורטו להלן.

שימו לב כי רצוי לעצב את הקוד בצורה המיטביית בלי שום קשר לדרישות בתרגיל זה, ובהחלט ניתן להשתמש לשם כך בתבניות עיצוב נוספות.

יינתן בונוס למערכות מעוצבות היטב שינצלו למקסימום את הרעיון של תבניות עיצוב (אך מבלי להתעלל במונח רק על מנת להשיג את הבונוס!).

**תבניות העיצוב שיש למממש:**

1. Singleton

יש למשש זאת במספר מקומות, לפחות בחלון שמהווה את המסלול נסיעה וכן בחלון הinfo.

1. Factory method

לשם יצירת כלי רכב.

1. Decorator

המימוש של צבע כלי הרכב וכן האורות צריכים להתבצע על ידי שימוש בתבנית העיצוב decorator.

ויש להוסיף אלמנט קישוטי נוסף של גבול (boarder) כל מכונית תוכל להיות עם או בלי (עם יגרור שבשרטוט המכונית יש גם לשרטט גבול מסביב התמונה).

ההחלטה האם יהיה גבול או לא למכונית תתבצע על ידי הוספה של כפתור לGUI לאחר לחיצה עליו ניתן לבחור אחד מ-5 כלי הרכב הקיימים – ואז אם לאותו כלי הרכב לא היה גבול – הוא יתווסף עכשיו, ואם היה גבול – הוא יבוטל.

1. Observer
2. במקרה של סגירת המערכת יהיה שימוש ב observer במקום ב-interrupt על מנת לאפשר סגירה מסודרת של כל המכוניות ורק אז סגירת החלון.
3. אם לאחת מהמכוניות יסתיים הדלק הכפתור בGUI שמאפשר מילוי דלק ישנה את צבעו לאדום. וזה יבוצע על ידי שימוש observer.
4. חלון info יאזין לשינויים במצבי המכוניות ויתעדכן בכל פעם שיש שינוי.
5. Thread pool – כמובן שכל ניהול המכוניות צרך להיות מבוצע על ידי כך שמכונית היא runnable ולא thread. יש תמיד 5 threads שעובדים וכל אחד מהם אחראי על מכונית (או בהמתנה אם אין מכוניות) – לפי כללי תבנית העיצוב.

**הערה:**  אתם יכולים להשתמש בתבנית הבנויה כבר בjava או לממש בעצמכם. יש אתגרים שונים בכל אחת משתי האפשרויות.

1. Memnto

יש להוסיף שורה נוספת של לחצים לGUI למאפשרים שמירת מצב memnto וכן שחזור של מצב memnto שמור.

שימו לב כי ישנם אפשרויות רבות נוספות לשימוש בתבניות עיצוב בתרגיל זה.

**עבודה נעימה!!!**